REMARKS

Reconsideration and allowance of the present application are respectfully requested. Claims 1-2, 4-15, 17-19, 30-34, 37 and 38 are currently pending in this application.

Regarding the 35 U.S.C. § 112, paragraph one Rejection

Claims 37-38 stand rejected under 35 U.S.C. §112, paragraph one. Applicant traverses the rejection. Support for Claims 37-38 may be found at least at page 30, lines 4-12; page 34, lines 8-18; page 36, lines 21-24 of the specification, which describes a system for generating forms inependently and prior to user interaction with the computer program.

Regarding the 35 U.S.C. § 102 Rejection

Claims 1, 2, 4-15, 17-19, and 30-34 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Published Patent Application No. 2004/0039993 to Kougiouris et al. (Hereinafter, "Kougiouris"). Applicant respectfully traverses this rejection for at least the following reasons.

Applicant respectfully disagrees with the Office's reading of the Kougiouris reference for at least the following reasons. For the immediate purposes, Applicant's comments are directed to Claim 1. Claim 1, in part, recites:

- "automatically identifying a set of one or more attributes of the computer program with values that are to be input to the computer program by a user; and

- creating code for one or more forms including selected ones of the set of one or more attributes."

The Office has cited Kougiouris, and specifically, paragraphs [0070]-[0075] for this disclosure. Applicant disagrees.

A chief difference of opinion between the Office and the Applicant is whether Kougiouris discloses "creating code". In the Final Action, the Office asserts that the disclosure of "The graphical user interfaces can be created from markup languages such as HTML or XML derived markup language descriptions" (Final Action Page 4, first paragraph) discloses "creating code". Applicant disagrees. The cited portion of Kougiouris discloses that ". . . the application program displays the GUI described in the markup language file." Kougiouris, Page 5, paragraph [0069]. Thus, while a markup language may be used (for ultimate display), the file (including the code to generate the GUI) already exists in the markup language file. This is to say that the purpose of the markup language file is to "describe" the GUI in code, such that the GUI may be displayed.

If the markup language file did not include code, then there would be no need to access the file, as the file would have nothing to add in order to display the

GUI as the code would be "created" rather than "presented". Claim 1 recites that "code for one or more forms" may be created. The Office's position is incorrect because Claim 1 recites "creating code", thus it is the code (which underlies the form) that is created and not the **display** or the expression of the code (using a particular markup language) as contended by the Office at page 9. During presentation in Kougiouris, the code is simply expressed. This is to say, that in Kougiouris the code existed before presentation or there would be no way to present the GUI. Thus, while XML may be used to generate the GUI in Kougiouris, this is not the question, rather the question is whether the code existed. This is to say that the markup language in and of itself does not create the code but merely formats the display of the code which is contained in the markup language file. In Kougiouris, the code is included in the markup language file as a result the GUI may be formed when the XML file is accessed. If the Office's position were correct, then there is no need for this file as the code would be "created" by the application of the markup language.

Additionally, Kougiouris fails to disclose "automatically identifying a set of one or more attributes of a computer program with values that are *to be* input to the computer program by a user." Emphasis added. The Office is incorrect that Kougiouris discloses this feature. Kougiouris fails to disclose this feature because Kougiouris is not directed to "creating code" but to presentation or display of a GUI which is already described by a markup language file. Thus, Kougiouris is not concerned with values "to be" included as the format of the entry is previously

been defined. Rather, Kougiouris merely confirms that the entry in the field matches the defined format (e.g., a social security number is nine digits long, contains only digits, and is expressed as xxx-xx-xxxx, rather than "automatically identifying a set of one or more attributes of the computer program with values that are *to be* input". For at least the foregoing reasons, the pending rejection is improper. Removal of the pending rejection is requested and allowance is solicited.

As the Office has not specifically rejected the language recited in independent Claim 30, Applicant traverses the rejection under a similar rationale as discussed above with respect to Claim 1, although differences exist between Claims 1 and 30. In general, Kougiouris fails to disclose, "automatically generating code" or to "analyzing the identified operations". Kougiouris fails to do this because Kougiouris is directed to a method for displaying a GUI rather than generating code and analyzing identified operations. In light of the foregoing, removal of the pending rejection is requested and allowance is solicited.

Claims 2 and 4-10 are allowable as depending from an independent claim which is in a condition for allowance. Applicant traverses the rejection. Claims 2, and 4-10 recite additional features which are not found in the art of record. Removal of the pending rejection is requested and allowance is solicited.

With respect to **independent Claim 11**, the pending rejection under 35 U.S.C. §102(e) is improper and should be removed for at least the following reasons. Claim 11, in part, recites:

- "identifying, for each of the command definitions of each of the plurality of interactions, the methods of the command definition,

- checking, for each identified method that sets a value, whether a corresponding identified method obtains the value, and

- identifying, as an attribute of the set of one or more attributes, each attribute corresponding to a method that sets a value for the attribute for which there is no corresponding identified method that obtains the value for the attribute;"

Kougiouris does not disclose these features because Kougiouris is directed to presenting a GUI which has been described in a markup language file rather than "checking, for each identified method that sets a value, whether a corresponding identified method obtains the value." In other words, the reference does not check to see if the method "obtains the value" but rather confirms that the text the user has entered is valid. (See below.)

[0117] For example, although user interfaces are discussed above with reference to markup language GUI descriptions, user interfaces may also be created in other ways, e.g., by a Visual Basic program. When a user enters text into a user interface text field created from a Visual Basic program and then tabs away from the control, the program may intercept an event indicating that the user tabbed away from the control, similarly to the description above. The program may then call a component method to verify that the text entered by the user is valid text, e.g. by verifying that the text is of a specific length or matches a particular pattern of characters. The application may then take appropriate action, e.g. by highlighting the user interface control to indicate that the text is invalid, or by calling another component method to format the valid text by inserting parentheses, dashes, etc. and then redisplaying the text in the user interface, etc.

Kougiouris, Page 8, paragraph [0117].

This fails to disclose checking if the "corresponding identified method obtains the value" but instead discloses seeing if the input is valid (e.g., a social security number is all numbers and nine digits long). Thus, while the claim is directed to checking a method (such as to determine if the method obtains the value) the reference is directed at checking entered text or data. As such, the reference fails to disclose each and every feature, and a *prima facie* case of anticipation does not exist. For at least the foregoing reasons, the pending 35 U.S.C. §102(e) rejection is improper. Removal of the pending rejection is requested and allowance is solicited.

Claims 12 and 13 are allowable as depending from independent Claim 11 which is in a condition for allowance. Applicant traverses the rejection. Claims 12 and 13 recite additional features which are not found in the art of record. Removal of the pending rejection is requested and allowance is solicited.

Applicant traverses the rejection of independent **Claim 14** for at least the following reasons. Claim 14, in part, recites:

- "accessing a computer program;

- automatically identifying a set of one or more outputs of the computer program;

- generating a list identifying the set of one or more outputs; and

- outputting the list,

wherein the identifying and generating are performed based on an analysis of computer program code, independent of execution of the computer program to provide one or more views."

The Office is incorrect that Kougiouris discloses each and every feature. In particular, as discussed is previous responses, Kougiouris fails to disclose "wherein the identifying and generating are performed based on an analysis of computer program code, independent of execution of the computer program to provide one or more views." The Kougiouris reference fails to disclose this feature as the Kougiouris process is not independent of the execution of the computer program.

In general, the Application describes a method in which a computer program may be tested or problems may be identified independently of operation. Application, page 31, lines 4-28. In particular, the application discloses that the code included in a tested program may be analyzed independent of executing the

program. For example, the code, to be included in a computer program, may be analyzed to determine how the definition would operate. Thus, a definition may be queried to determine operation without having to invoke the program using test parameters. In this fashion, the computer program does not need to be run or executed to identify the output of the code to "provide one or more views". In comparison, Kougiouris merely identifies if the input text is valid while the application generating the GUI is running. In other words, while Claim 14 generally recites "wherein the identifying and generating are performed based on an analysis of computer program code, independent of execution of the computer program to provide one or more views" Kougiouris requires execution of the GUI. For at least the foregoing reasons, the pending rejection is improper. Removal of the pending rejection is requested and allowance is solicited.

Claims 15 and 17-19 are allowable as depending from an independent claim which is in a condition for allowance. Applicant traverses the rejection. Claims 15 and 17-19 recite additional features which are not found in the art of record.

For at least the above-identified reasons, the Applicant submits that the 35 U.S.C. § 102(e) rejection is improper, and therefore respectfully requests that it be withdrawn.
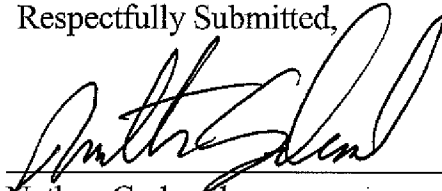
Conclusion

The arguments presented above are not exhaustive; Applicant reserves the right to present additional arguments to fortify its position. Further, Applicant reserves the right to challenge the alleged prior art status of one or more documents cited in the Office Action.

In conclusion, all objections and rejections raised in the Office Action having been addressed, it is respectfully submitted that the present application is in condition for allowance and such allowance is respectfully solicited. The Examiner is urged to contact the undersigned if any issues remain unresolved by this Amendment.

Respectfully Submitted,

Dated: 8.16.07   By: _____

Nathan Grebasch
Reg. No. 48,600
(509) 324-9256